The Challenge of Energy-Efficient HPC

The electrical power demands of ultrascale computers threaten to limit the future growth of computational science. To reach exascale computing cost-effectively, a group of researchers propose to radically change the relationship between machines and applications by developing a tightly-coupled hardware/software co-design process. The Green Flash project is intended to dramatically accelerate the development cycle for exascale systems while decreasing the power requirements.

The electrical power demands of ultrascale computers threaten to limit the future growth of computational science.

In April, May, and June 2007, three town hall meetings were held at Lawrence Berkeley, Oak Ridge, and Argonne national laboratories to collect community input on the prospects of a proposed new DOE initiative entitled Simulation and Modeling at the Exascale for Energy and the Environment, or E3 for short. About 450 researchers from universities, national laboratories, and U.S. companies discussed the potential benefits of advanced computing at the exascale (10^{18} operations per second) on global challenge problems in the areas of energy, the environment, and basic science. The findings of the meetings were summarized in a document that quickly became known as the E3 Report.

The E3 Report stated that exascale computer systems are expected to be technologically feasible within the next 15 years, but that they face significant challenges. One of the challenges receiving a great deal of attention throughout the high-performance computing (HPC) community is power efficiency. An exaflop/s system that requires less than 20 megawatts (MW) of sustained power consumption (enough to power approximately 2,600 homes) is "perhaps achievable," according to the E3 findings, if computers become more power efficient. But if existing technology is simply extrapolated into the future, power estimates grow roughly an order of magnitude higher. When the cost of electricity to run and cool a supercomputer grows to exceed its procurement cost (which is already happening at some major data centers), the economic viability of such projects may come into question.

The electrical power demands of ultrascale computers threaten to limit the future growth of computational science. For decades, the notion of computer performance has been synonymous with raw speed as measured in flop/s (floating point operations per second). That isolated focus has led to supercomputers that consume egregious amounts of electrical power. Other performance metrics—such as power efficiency, space efficiency, reliability, availability, and usability—have been largely ignored. As a consequence, the total cost of ownership of a supercomputer has increased extraordinarily. The current approach to building supercomputers is not sustainable without dramatic increases in funds to operate the systems.

While Moore's Law—which predicts that the number of transistors per chip will double every 18 months—is alive and well, more transistors are no longer resulting in faster chips that consume less energy. Traditional methods for extracting more performance per processor have been well mined. The only way to improve performance now is to put more cores on a chip. In fact, it is now the number of cores per chip that is doubling every 18 months, instead of clock frequency doubling, as it has in the past.

Consequently, the path towards realizing exascale computing depends on riding a wave of exponentially increasing system concurrency, in which tens to hundreds of processes are executing at the same time—not just within the entire system, as in massively parallel computing, but within each multiple-core processor. This is leading to reconsideration of interconnect design, memory balance, and input/output (I/O) system design. The entire software infrastructure is built on assumptions that are no longer true. The shift to multicore (two, four, or eight cores) and manycore processors (tens to hundreds of cores) will have dramatic consequences for the design of future HPC applications and algorithms.

To reach exascale computing cost-effectively, a group of researchers from the National Energy Research Scientific Computing Center (NERSC) Division and the Computational Research Division (CRD) at Lawrence Berkeley National Laboratory (LBNL, or Berkeley Lab) proposed to radically change the relationship between machines and applications by developing a tightly-coupled hardware/software co-design process.

Their project—named Green Flash after the optical phenomenon that sometimes appears on the horizon at sunset or sunrise—has the aggressive goal of achieving 100 times the computational efficiency and 100 times the capability of the mainstream HPC approach to hardware/software design. We propose to use global cloud system resolving models for climate change simulation as one of the key driver applications to develop the hardware/software co-design methodology. This hardware/software co-design process is intended to dramatically accelerate the development cycle for exascale systems while decreasing the power requirements.

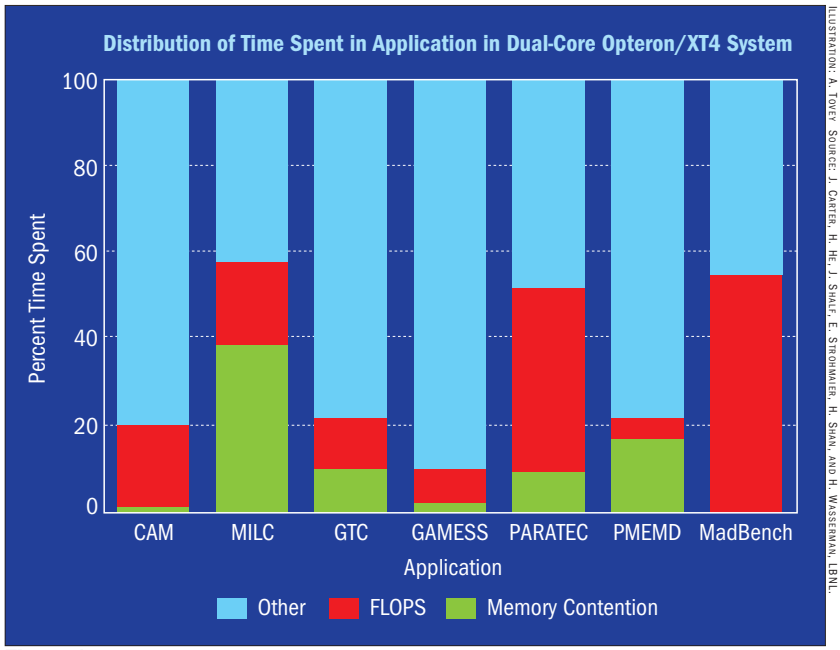


ILLUSTRATION: A. TOVEY SOURCE: J. CARTER, H. HE, J. SHALF, E. STROHMAIER, H. SHAN, AND H. WASSERMAN, LBNL.

Figure 1. *A breakdown of where time was spent in a subset of the NERSC SSP application codes suggests that different applications have different requirements for computational efficiency.*

Reducing Waste in Computing

The low-power, embedded-computing market—including consumer electronics products such as cell phones, PDAs, and MP3 players—has been the driver for processor innovation in recent years. The processors in these products are optimized for low power (to lengthen battery life), low cost, and high computational efficiency.

According to Mark Horowitz, Professor of Electrical Engineering and Computer Science at Stanford University and co-founder of Rambus Inc., "Years of research in low-power embedded computing have shown only one design technique to reduce power: reduce waste." The sources of waste in current HPC systems include wasted transistors (surface area), wasted computation (useless work, speculation, stalls), wasted bandwidth (data movement), and chip designs optimized for serial performance, which increases the complexity (and power waste) of the design.

Efficient designs must be specific to application and/or algorithm classes, as suggested by a NERSC/CRD study that examined the dual-core AMD processor used in the Cray XT3 and XT4 systems to assess the current state of system balance and to determine when to invest more resources to improve memory bandwidth. The study used the NERSC SSP benchmark, which is a diverse array of full-scale applications that represent a significant fraction of the NERSC workload. A breakdown of time spent in various components of the codes shows that surprisingly little time could be attributed to memory contention corresponding to basic memory bandwidth limitations (figure 1). The largest fraction of time (the "Other" category) is attributed to either latency stalls or integer/address arithmetic. Theoretically, these applications should all be memory-bandwidth bound, but instead the study shows that most are constrained by other microarchitectural bottlenecks in existing processors, and that different applications have different balance requirements.

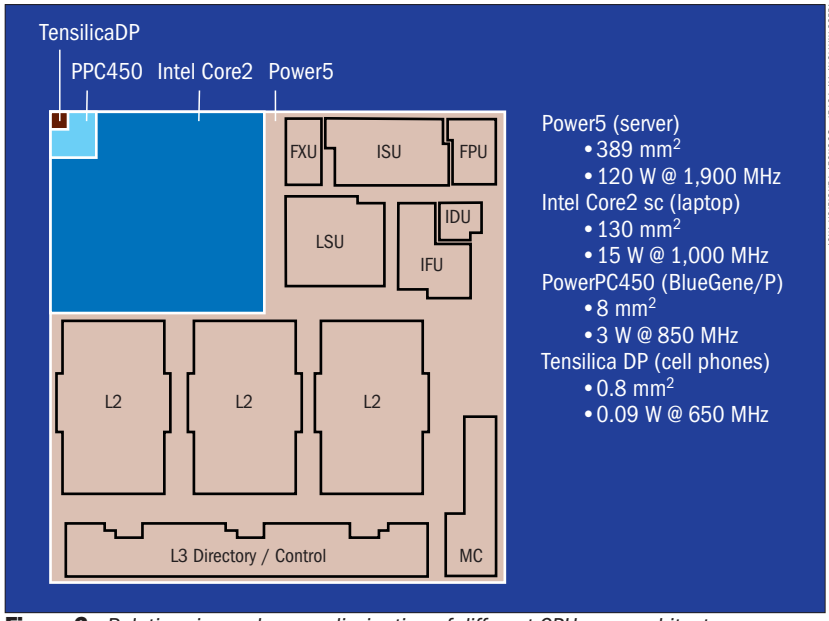


ILLUSTRATION: A. TOVEY SOURCE: TENSILICA INC.

Figure 2. *Relative size and power dissipation of different CPU core architectures. Simpler processor cores require far less surface area and power with only a modest drop in clock frequency. Even if measured by sustained performance on applications, the power efficiency and performance per unit area is significantly better when using the simpler cores.*

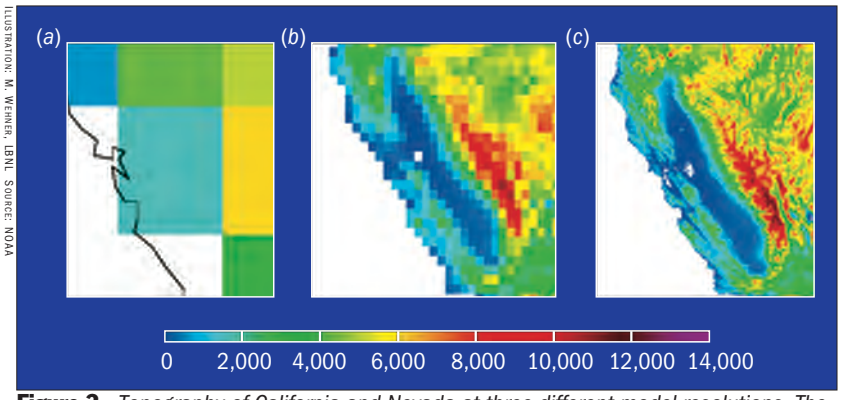
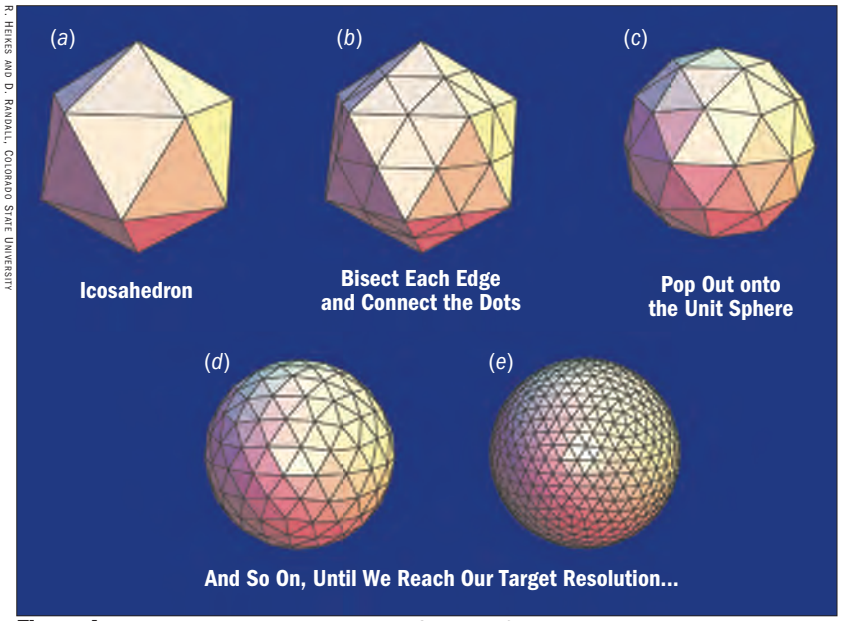


ILLUSTRATION: M. WEHNER, LBNL SOURCE: NOAA

Figure 3. *Topography of California and Nevada at three different model resolutions. The left panel shows the relatively low resolution typical of the models used for the International Panel on Climate Change's Fourth Assessment Report, published in 2006. The center panel shows the upper limit of current climate models with statistical approximations of cloud systems. The right panel shows the resolution needed for direct numerical simulation of individual cloud systems.*



R. HEIKES AND D. RANDALL, COLORADO STATE UNIVERSITY

Figure 4. *The geodesic mesh used by the Colorado State University group to represent the Earth's atmosphere is generated from an icosahedron (a). In this scheme, the triangular faces of the icosahedron are first bisected (b), then the new vertices are projected onto a sphere (c), as if it were a ball being inflated. This procedure is repeated (d, e) until the desired resolution is obtained. The Green Flash target resolution is 167,772,162 vertices.*

A core designed to a specific set of application resource requirements can get 10–100 times better performance per watt, as shown by studies from Stanford University and from Tensilica Inc. Figure 2 (p51) illustrates this potential by showing the area and performance differences between general purpose, embedded (used in IBM's BlueGene/P), and application-tailored cores. The figure shows how much area and power desktop processors waste because they are optimized for serial code. The DOE's science applications, because they are already highly parallel, are an excellent driver for understanding how processors can be designed to optimize for efficient parallel execution rather than serial execution.

Parallelism is an energy-efficient way to achieve performance. A system with many simple cores offers higher performance per unit area for parallel codes than a comparable design employing smaller numbers of complex cores. Lower complexity makes a chip more economical to design and produce, and smaller processing elements provide an economical way to improve defect tolerance by providing many redundant cores that can be turned off if there are defects.

Figure 2 (p51) shows that moving to a simpler core design results in modestly lower clock frequencies, but has enormous benefits in chip surface area and power consumption. Even if it is assumed that the simpler core will offer only one-third the computational efficiency of the more complex out-of-order cores, a manycore design could still provide an order of magnitude more power efficiency for an equivalent sustained performance. As the figure illustrates, even with the smaller cores operating at one-third to one-tenth the efficiency of the largest chip, 100 times more cores can still be packed onto a chip and consume one-twentieth the power. Effective performance per watt is the critical metric.

This design approach brings with it an even greater challenge: creating ultrascale parallel applications that can run effectively on this radically different architecture.

A Hardware/Software Co-Design Process

If the HPC community emulated the embedded computing industry, we could potentially reduce not only power requirements but also design costs and time to market. A key limiting factor in the market-driven approach to HPC procurements is the length of the feedback loop on system designs. Due to the high design investment cost, the vendor must make compromises in the system design to accommodate a wide variety of applications. The application scientists cannot provide performance feedback to the vendor until hardware is released for testing and evaluation. This multi-year cycle is a source of significant inefficiencies for scientific productivity, because it can take years for each new iteration of hardware to become available for testing and evaluation by the application scientists. A hardware/software co-design approach could dramatically accelerate this process.

For years, NERSC has engaged in a cooperative effort with hardware designers called Science-Driven System Architecture, which involves engaging application scientists in the early stages of the hardware design process for future-generation supercomputing systems. The Green Flash team is

The Green Flash team is currently focusing this cooperative effort toward a new design paradigm: application-driven HPC.

Global Cloud Resolving Models

While global warming is now considered unequivocal by the leading national and international scientific organizations, many uncertainties remain in the details of future climate change. Principal among the uncertainties of the natural portion of the climate system is the role of clouds.

Clouds act in many complex ways to influence the transport of moisture and energy within the atmosphere. Cumulus convective clouds, especially in the tropics, transport moisture from the lower atmosphere to higher altitudes where it is carried by winds to other parts of the globe. Clouds affect the radiative energy budget by reflecting sunlight from above and trapping infrared radiation from below.

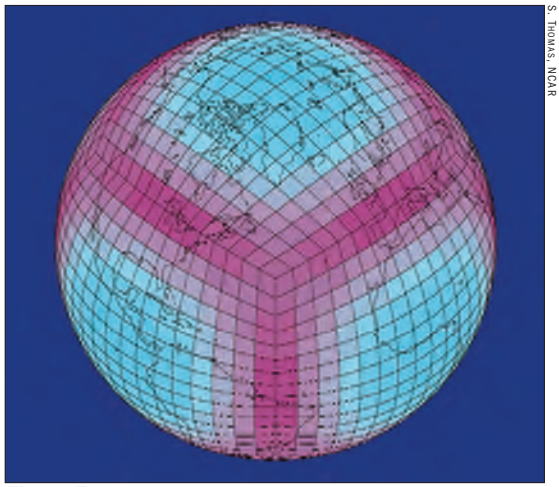
Current generation climate models cannot resolve individual clouds because the horizontal resolution is far too coarse (figure 3). In these models, the effect of clouds is parameterized. In cloud parameterizations, the statistical properties of clouds are modeled, usually in an *ad hoc* manner. However, current climate models using this approach typically produce very poor simulations of cloud distributions when compared to observations. This results in biases in the pattern of precipitation as well as the radiation budget.

An alternative approach is a direct numerical simulation (DNS) of cloud systems, which would be possible if computers powerful enough to permit atmospheric models of resolutions approaching 1 km could be built. In these global cloud resolving models (GCRMs), cloud systems (not individual clouds) are simulated directly. Such models are far closer to the first principles governing clouds than are the current statistical parameterizations, and they offer genuine promise to reduce climate model errors.

For the Green Flash project, we initially estimated computational requirements by extrapolating measurements from fvCAM, the Community Atmospheric Model developed at the National Center for Atmospheric Research (NCAR) in Boulder, Colorado. However, because of convergence of the mesh at the poles, this code cannot practically be extended to these high resolutions. To make more credible estimates, we are repeating our original study with the spherical geodesic grid-based atmospheric model developed with SciDAC support at Colorado State University (figure 4). The CSU code is only one of several advanced grid technologies that would permit resolutions of order 1 km. Another advanced mesh capable of this resolution is the cubed-sphere grid (figure 5), undergoing incorporation in new atmospheric models at both the National Oceanic and Atmospheric Administration (NOAA) and the National Aeronautics and Space Administration (NASA). It is well established that results from multiple, independent models improve the credibility of future climate projections, hence any architecture developed in the Green Flash project would need to be flexible enough to efficiently integrate codes using both the geodesic and cubed-sphere approaches. To be of greatest use, architecture design parameters are best targeted towards a class of models rather than any individual model. For GCRMs, this approach is a sound one as the differences in computational requirements between climate models is much smaller than the range of computational requirements across the breadth of DOE Office of Science exascale applications.

currently focusing this cooperative effort toward a new design paradigm: application-driven HPC. This approach involves identifying high-impact exascale scientific applications, tailoring the system architecture to the application resource requirements, and co-designing algorithms and software together with the semi-custom hardware.

The first application to be chosen for the approach is the geodesic global cloud-resolving model (GCRM) being developed by David Randall and his group at Colorado State University (CSU), where he is Director of the Center for Multiscale Modeling of Atmospheric Processes and principal investigator of the DOE SciDAC project "Design and Testing of a Global Cloud-Resolving Model" (sidebar "Global Cloud Resolving Models"). Although the current SciDAC project aims to develop a cloud model with a 3 km horizontal grid resolution, the ultimate goal is 1 km resolution, which would allow researchers to replace the statistical approximations of cumulus convective cloud systems used in current climate models with direct numerical simulations of individual cloud systems, providing more precise modeling of heat and moisture transport (figure 3).



S. Thomas, NCAR

Figure 5. *The cubed-sphere grid provides a quasi-uniform mapping of grid cells on the sphere for solving the equations of motion, thermodynamics, and moist physics within the atmosphere.*

To develop a 1 km cloud model, scientists would need a supercomputer that is 1,000 times more powerful than what is available today. But building a supercomputer that powerful with conventional

To develop a 1 km cloud model, scientists would need a supercomputer that is 1,000 times more powerful than what is available today.

ILLUSTRATION: A. TOVEY SOURCE: J. SHALF, LBNL

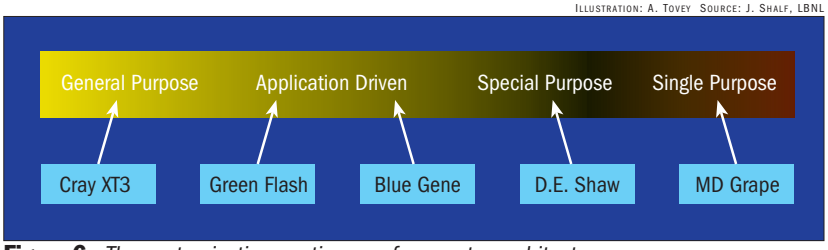


Figure 6. *The customization continuum of computer architectures.*

ILLUSTRATION: A. TOVEY SOURCE: J. SHALF, LBNL

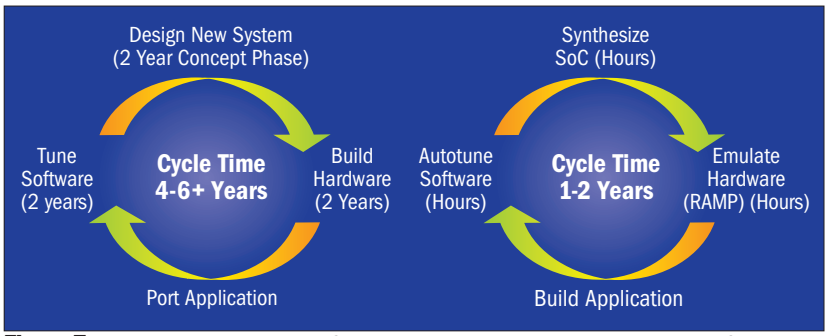


Figure 7. *The typical design cycle for computer systems takes multiple years from concept to completion. RAMP enables us to accelerate the feedback process for scientific application software to influence hardware designs by enabling us to modify and test architectural designs in hours rather than years.*

J. DAVIS AND C. THACKER, MICROSOFT RESEARCH

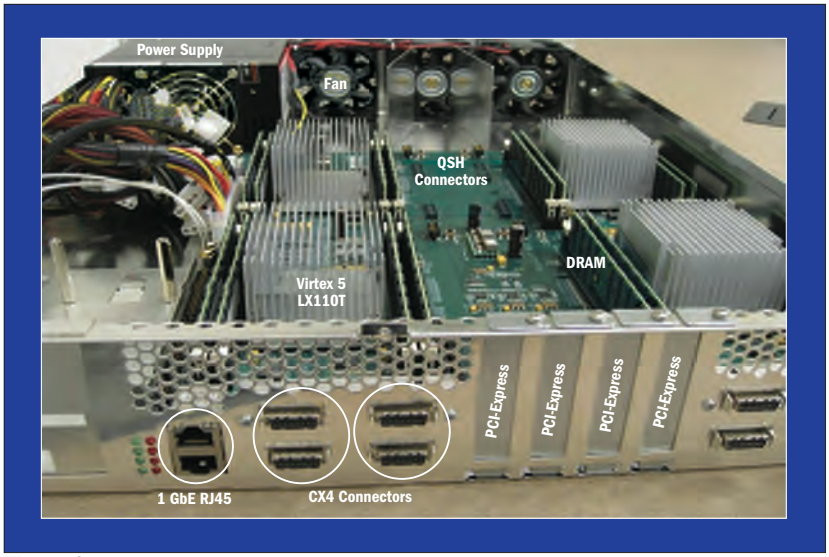


Figure 8. *The RAMP environment runs on a Berkeley BEE3 (Berkeley Emulation Engine 3) FPGA platform. The BEE3 has four Virtex-5 LX110T FPGAs that are tightly interconnected via RapidIO channels.*

microprocessors (the kind used to build personal computers) would cost about $1 billion and would require 200 megawatts of electricity to operate—enough energy to power a small city of 100,000 residents.

The Green Flash system will employ power-efficient cores specifically tailored to meet the requirements of this ultrascale climate code. The requirements of an atmospheric model at 1 km resolution are dominated by the equations of motion because the Courant stability condition requires smaller time steps. To be useful for projections of future climate change, the model must run at least 1,000 times faster than real time, calculating values for about two billion icosahedral points. At this rate, millennium-scale control runs could be completed in a year, and century-scale transient runs could be done in a month. The computational platform's sustained performance would need to reach at least 10 petaflop/s. This goal could be achieved with 20 million processors (consuming less than 4 MW of power), modest vertical parallelization, a modest speed of 0.5 gigaflop/s per processor, and 5 MB memory per processor.

An application-driven architecture does not necessitate a special-purpose machine, nor does it require exotic technology. As figure 6 shows with several examples, there is a customization continuum from general-purpose to single-purpose computers, and indeed the IBM Blue Gene line of systems was started with a very narrow application target in mind.

At the single-purpose, fully custom extreme is MD-Grape, a computer at RIKEN in Japan. MD-Grape was custom designed for molecular dynamics simulations (the name stands for "Molecular Dynamics-Greatly Reduced Array of Processor Elements") and has an application-specific integrated circuit (ASIC) chip design. It achieves 1 petaflop/s performance for its target application using 200 kilowatts of power, and cost $8.6 million from concept to implementation (including labor). Although MD-Grape was custom designed for molecular dynamics, it has proven useful for several other applications, including astrophysical N-body simulations.

An example of a semicustom design with some custom elements is the D. E. Shaw system. D. E. Shaw Research, a unit of an investment firm, focuses on the development of new algorithms and specialized supercomputer architectures for ultrafast biomolecular simulations of scientific and pharmaceutical problems. The D. E. Shaw system uses fully programmable cores with full-custom coprocessors to achieve efficiency, and simulates 100–1,000 times longer timescales than any other HPC system. While the programmability of the D. E. Shaw system broadens its application reach, it is still narrower than Berkeley Lab's Green Flash.

IBM's Blue Gene is the best example to date of the kind of application-driven architecture based on an embedded processor core that the Green Flash project envisions. Designed around a protein folding application, Blue Gene, over several generations, has proved to be useful for a growing list of applications, including hydrodynamics, quantum chemistry, molecular dynamics, climate modeling, and financial modeling.

Like Blue Gene, the Green Flash system will have a semicustom design. The core architecture

Like Blue Gene, the Green Flash system will have a semicustom design.

RAMP: Hardware Emulation for Rapid Prototyping

A key limiting factor in the application-driven approach is the length of the feedback loop on system designs (figure 7). The application scientists cannot provide feedback to the vendor until hardware is released for testing and evaluation; however, there is a long time lag between hardware iterations. Software simulation environments, which are typically used to evaluate system architecture during the early stages of the development process, are typically too slow to provide a credible testing environment for the target applications and often miss important behavioral details that only become manifest in the actual hardware infrastructure. There is therefore a critical need for a hardware emulation platform that is capable of simulating a system before its arrival, providing the lead time necessary to identify performance bottlenecks and develop robust software infrastructure.

To accelerate the feedback loop for prototype system designs, we are using the Research Accelerator for Multiple Processors (RAMP), an FPGA emulation platform that makes the hardware configuration available for evaluation while the actual hardware is still on the drawing board (figures 8 and 9). RAMP is a cooperative effort between six universities to build a new standard emulation system for parallel processors. By utilizing large field programmable gate arrays (FPGAs), RAMP looks like the real hardware to software developers, running at least 1,000 times faster than software simulators (despite being slower than the real hardware).

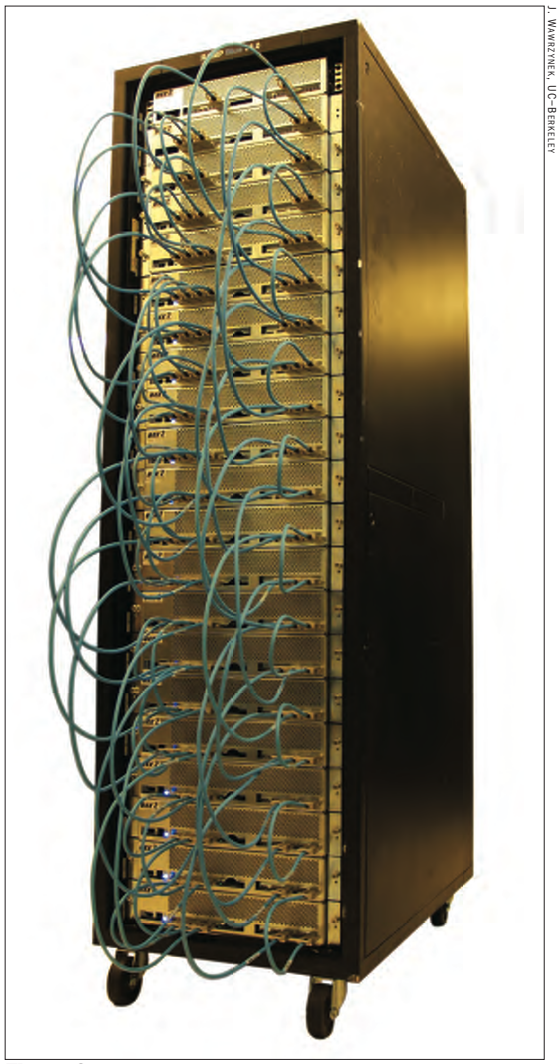
The flexibility of FPGAs allows RAMP-based systems to be constructed quickly and to be easily modified to experiment with different hardware alternatives, such as the number of processors, number of floating-point units per processor, size and speed of caches, prefetching schemes, speed of memory, and so on. With RAMP, one can generate a new system design ("tape out") every day, and the "build time" of a new system is a few minutes. This same process takes months or even years using conventional development processes.

will be highly programmable using C, C++, or Fortran. Its projected 100 times improvement in power efficiency would be modest when compared with the demonstrated capability of more specialized approaches. This approach will solve an exascale problem without building an exaflop/s machine.

The hardware/software co-design process would be impossible using the typical multi-year hardware lead times for complex, serial-optimized chips. However, a typical embedded processor vendor may generate up to 200 unique designs every year for simple, specialized chips. In order to keep up with the demand for semi-customized designs, leading embedded-design houses—such as IBM Microelectronics, Altera, and Tensilica—have evolved sophisticated toolsets to accelerate the design process through semi-automated synthesis of custom processor designs. We are now leveraging the expertise of this technology sector by collaborating with Mark Horowitz of Stanford University and Rambus, Inc., and Chris Rowen of Tensilica Inc.

Our co-design process utilizes the Research Accelerator for Multiple Processors (RAMP), an emulation platform that makes the hardware configuration available for evaluation while the actual hardware is still on the drawing board (sidebar "RAMP: Hardware Emulation for Rapid Prototyping"). The flexibility of RAMP allows rapid changes in the details of the hardware configuration, making hardware/software co-design feasible. In fact, we have recently demonstrated a 30%–50% improvement in power efficiency over conventional approaches to design space exploration.

The software side of the co-design process is supported by auto-tuning tools for code generation



J. WAWRZYNEK, UC–BERKELEY

Figure 9. *Multiple BEE3 boards can be interconnected into a rack-scale system using 10 Gigabit Ethernet to emulate much larger systems.*

Auto-Tuning for Chip Multiprocessors

The computing industry has recently moved away from exponential scaling of clock frequency toward chip multiprocessors (CMPs) in order to better manage trade-offs among performance, energy efficiency, and reliability. Because this design approach is relatively immature, there is a vast diversity of available architectural features and memory hierarchies making it extremely difficult to determine which CMP philosophy is best suited for a given class of algorithms. Likewise, this architectural diversity leads to uncertainty on how to refactor existing algorithms and tune them to take the maximum advantage of existing and emerging platforms. Understanding the most efficient design and utilization of these increasingly parallel multicore systems is one of the most challenging questions faced by the computing industry in several decades.

Auto-tuners have the potential to address these challenges. The idea behind auto-tuning is searching over a set of optimizations and their parameters to minimize runtime, while providing performance portability across the breadth of existing and future architectures. Auto-tuners augment compiler technology by enabling application-specific optimizations that could not be inferred automatically in generic program analysis. Given the complex and unpredictable interactions between an optimization and the underlying architecture, we believe that application-specific auto-tuners are the most practical near-term approach for consistently obtaining high performance on multicore systems. Previously auto-tuning has been successful for several key numerical kernels on serial processors, while today's auto-tuning efforts focus on the multicore regime.

Figure 10 presents an example of the types of blocking optimization choices that can be made for stencil (nearest-neighbor) computations—a class of algorithms at the heart of most calculations involving structured grids, including both implicit and explicit partial differential equation solvers. As shown in the figure, there are a myriad of decisions as to how most effectively to decompose the domain into core blocks, thread blocks, and register blocks such that performance is maximized for a given architectural instantiation. Auto-tuning allows the automation of this process, thus enabling programmer productivity as well as performance portability.

Figure 11 (left), shows the impact of auto-tuning a variety of optimization techniques for a Laplacian calculation on the AMD quad-core Opteron Barcelona—demonstrating performance improvements of up to 4.5 times compared with the naive implementation. Additionally, figure 11 (right) highlights that these techniques can be applied to a variety of platforms; in this case the Intel Clovertown, AMD Barcelona, Sun Victoria Falls, Cell QS22, and NVIDIA GTX280. By maximizing performance, auto-tuning allows for differentiation of power efficiency across the variety of architectures. Overall, we believe that auto-tuning will have significant impact on improving the performance, productivity, and portability of key numerical applications in the multicore era.

In addition to enabling a breakthrough in cloud-resolving climate simulation, Green Flash's power-efficient, application-driven design methodology will have an impact on the broader DOE scientific workload.

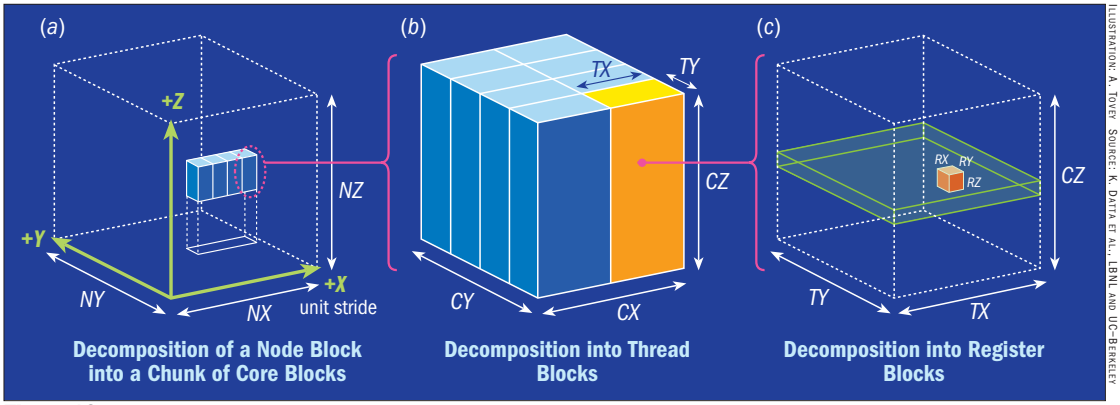


ILLUSTRATION: A. TOVEY SOURCE: K. DATTA ET AL., LBNL AND UC-BERKELEY

Figure 10. *Example of four-level problem decomposition for a stencil computation: from the original domain into core blocking, thread blocking, and register blocking. Auto-tuning automates the process of determining the optimal performance configuration for a given architectural instantiation, thus maximizing performance and programmer productivity.*

that are being developed by the SciDAC Center for Scalable Application Development Software (CScADS), led by John Mellor-Crummey of Rice University (sidebar "Auto-Tuning for Chip Multiprocessors").

In addition to enabling a breakthrough in cloud-resolving climate simulation, Green Flash's power-efficient, application-driven design methodology will have an impact on the broader DOE scientific workload. This hardware/software co-design approach is geared for a class of codes, not just for a single code instantiation, so this methodology is broadly applicable and could be extended to other scientific disciplines. Blue Gene was originally targeted at chemistry and bioinformatics applications, resulting in a power-efficient architecture, but its application has been broader than the original target. A similar result is expected from Green Flash.

First Prototype Runs Successfully

The Green Flash project has successfully reached its first milestone by running a next-generation,

ILLUSTRATION: A. TOVEY. SOURCE: K. DATTA ET AL., LBNL AND UC–BERKELEY

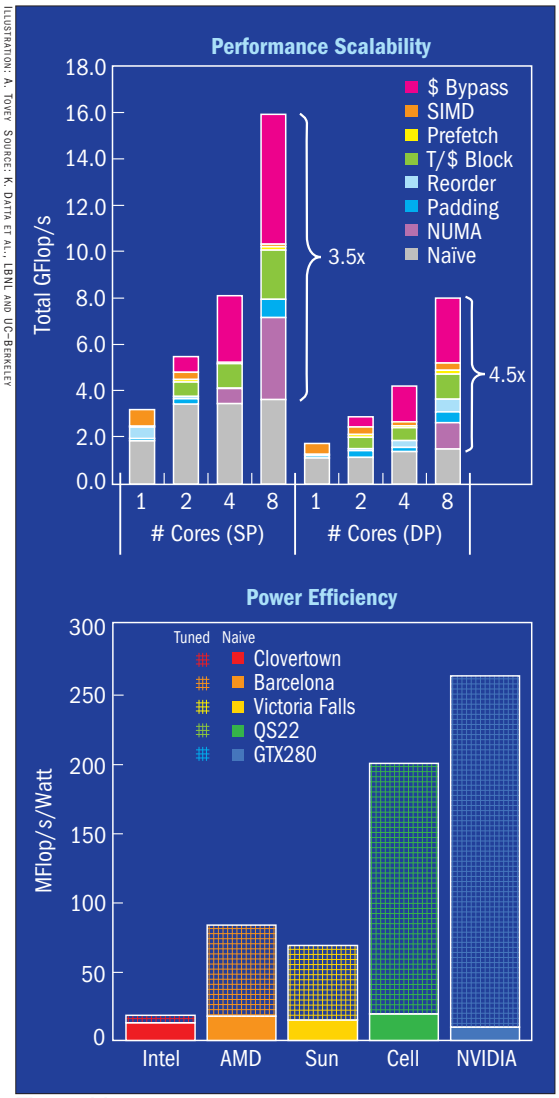


Figure 11. *The impact of auto-tuning showing (top) performance improvement of a Laplacian stencil computation on the AMD Barcelona, showing speedup of up to 4.5 times compared with the naive version, and (bottom) portability across a broad range of architectural platforms, demonstrating the effect of optimized performance on power efficiency.*

limited area model version of the GCRM on a logical prototype of a Green Flash processor. The logical prototype, which simulates the entire circuit design of the proposed processor, was designed in collaboration with Tensilica, using Tensilica's Xtensa LX extensible processor core as the basic building block, and was run on a RAMP BEE3 hardware emulator. A demonstration of the prototype ran at the SC08 conference in Austin, Texas in November 2008 (figure 12).

The hardware goals for the prototype research are fairly simple: produce a hardware prototype of a single Green Flash processor by fall 2009, and an entire node of the system (64 to 128 processors) by fall 2010. But the software goals are more challenging.

N. KEEN, LBNL

Figure 12. *David Donofrio of LBNL demonstrates the Colorado State University limited-area geodesic climate model running on RAMP at Supercomputing 2008. The RAMP BEE3 board (in the foreground) is running a cycle-accurate circuit-level simulation of the science-application-optimized embedded processor core that would be used in the Green Flash system. The simulation runs at 100 MHz, which is only five times slower than the target clock frequency of 500 MHz. RAMP enables early evaluation of systems using full applications, whereas software simulation forces researchers to make design decisions based on small extracted kernels.*

There are open issues that we are still wrestling with. We know how to program embedded processors for the GCRM code, but we have not fully explored the more general question of how to program a machine with 20 million processors. But by examining this one particular case, we hope to learn lessons that can be generalized to other codes with similar algorithms on a machine of this scale. Whether or not a full-scale Green Flash system is built, the answers to these questions will be the most important research challenges for the computer science community for the next decade. ●

Contributors Dr. John Shalf, NERSC Division, LBNL; Dr. Michael Wehner and Dr. Leonid Oliker, Computational Research Division, LBNL; John Hules, Computing Sciences Communications, LBNL

Acknowledgements

Green Flash is funded by the LBNL Laboratory Directed Research and Development program. All authors from LBNL were supported by the Office of Advanced Scientific Computing Research or the Climate Change Prediction Program under the Office of Biological and Environmental Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231

Further Reading

Green Flash
http://www.lbl.gov/CS/html/greenflash.html